UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/681,519	04/22/2001	Derrick T. Babb		4627

28686 7590 03/18/2005

DERRICK T. BABB
751 EVANGELINE AVENUE
ORLANDO, FL 32809

EXAMINER
WINDER, PATRICE L

ART UNIT	PAPER NUMBER
2145	

DATE MAILED: 03/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

TC2100 PK2

Organization _____ Bldg./Room _____

U. S. DEPARTMENT OF COMMERCE
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
IF UNDELIVERABLE RETURN IN TEN DAYS

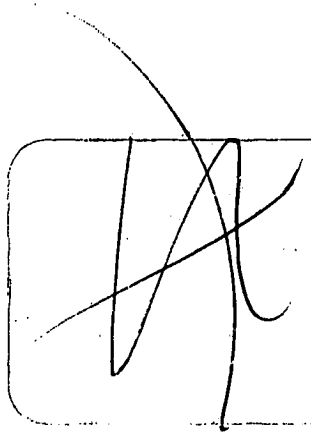
OFFICIAL BUSINESS

Y 3/2 3/3

Unable As Addressed
Unable To Forward
Insufficient Address
Moved, Left No Address
Unclaimed Refused
Attempted-Not Known
No Such Street Number
Vacant Illegible
No Mail Receptacle
Box Closed-No Order
Returned For Better Address
Postage Due

UNITED STATES POSTAGE
02 1A
0004204055
MAILED FROM

AN EQUAL OPPORTUNITY EMPLOYER



Office Action Summary	Application No.	Applicant(s)	
	09/681,519	BABB, DERRICK T.	
	Examiner	Art Unit	
	Patrice Winder	2145	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 April 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1* is/are rejected.

7)☒ Claim(s) *1* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

re

DETAILED ACTION

Claim Objections

1. Claim 1 is objected to because of the following informalities: improper form, i.e. more than one sentence indicated by more than one period. A properly written claim is a single sentence. Appropriate correction is required.

2. Claim 1 is objected to because of the following informalities: "I claim" is recited in the claim language. "I claim" can be a sub-heading under the title "Claims" but must not be recited in the claim itself. Appropriate correction is required.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claim 1 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Applicant's specification lacks description of the "Matrix Network Control System". The specification does not provides no insight as to network the Matrix Naming Structure is incorporated into or the control system that coordinates interaction with the Matrix Naming Structure.

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claim 1 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite in that it fails to point out what is included or excluded by the claim language. This claim is an omnibus type claim. The following language renders the claim 1 an omnibus type claim "outlined in this patent [application] as my invention".

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

9. Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ryan, USPN 6,412,014 B1 (hereafter referred to as Ryan) in view of Weider et al., USPN 6,374,253 B1 (hereafter referred to as Weider).

10. Regarding claim 1, Ryan taught a Matrix Network Control System and Matrix Naming Structure outlined in this patent application (abstract), the matrix naming structure (directory 7) specifically includes creating a matrix (dimension x = type of activity, dimension y = entity, column 5, lines 11-19), a collective structure containing sub-objects, that is represented by number(s), letter(s), or an alpha-numeric combination (directory 7, column 5, lines 20-23), separated by a space (separate by the space between lines), then combined with an identifying name, number, or alpha-numeric combination to form a root definition (column 5, lines 38-45);

wherein the naming structure also includes developing branches (column 4, lines 37-49) and forming aliases (column 5, lines 23-30); and

wherein Matrix Control System specifically includes developing a network structure using the Matrix Naming Structure (column 4, line 63 – column 5, line 5), including the development of a Global Matrix (column 3, lines 52-61). Ryan does not specifically the using delimiters "|" character and "@" character. However, Weider taught using the delimiters "|" and "@" (column 7, lines 22-28). It would have been obvious to one of ordinary skill in the art at the time the invention was made that incorporating Weider's delimiters in Ryan's Internet Directory System would have provided Ryan's system greater flexibility. The motivation would have been to increase the number of characters which can be prescribed delimiters.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Belissent et al., USPN 6,685,594 B1: taught when a message having a new domain name is received, a directory server creates a corresponding entry in a directory for every component that does not already exist in the directory;

b. William Wong, Accessing the Directory Database Using Distinguished Names: taught a Distinguished Name (DN) is a list of items, separated by commas, where each item is an attribute name followed by an equals sign (=) and a value; and

c. Timothy A. Howes, The Lightweight Directory Access Protocol: X.500 Lite: taught that in LDAP and X.500 directory services entries are arranged in a tree structure and divided among servers in a geographical and organizational distribution. Entries are named according to their position in this hierarchy by a distinguished name (DN).

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Patrice Winder whose telephone number is 571-272-3935. The examiner can normally be reached on Monday-Friday, 10:30 am-7:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Valencia Martin-Wallace can be reached on 571-272-6159. The fax phone

number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Patrice Winder
Primary Examiner
Art Unit 2145

March 11, 2005

Notice of References Cited	Application/Control No. 09/681,519	Applicant(s)/Patent Under Reexamination BABB, DERRICK T.	
	Examiner Patrice Winder	Art Unit 2145	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,374,253 B1	04-2002	Weider et al.	707/102
	B	US-6,412,014 B1	06-2002	Ryan, William Kenneth	709/245
	C	US-6,865,594 B1	03-2005	Belissent et al.	709/238
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	William Wong, Accesing the Directory Database Using Distinguished Names, WindowIT Pro: Connecting the IT Community, March 1999, InstantDoc #4901, 3 pages
	V	Timothy A. However, The Ligthweight Directory Access Protocol: X.500 Lite, CITI Technical Report 95-8, University of Michiga July 1995, 7 pages
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

CITI Technical Report 95-8

The Lightweight Directory Access Protocol: X.500 Lite

Timothy A. Howes
tim@umich.edu

ABSTRACT

This paper describes the Lightweight Directory Access Protocol (LDAP), which provides low-overhead access to the X.500 directory. LDAP includes a subset of full X.500 functionality. It runs directly over TCP and uses a simplified data representation for many protocol elements. These simplifications make LDAP clients smaller, faster, and easier to implement than full X.500 clients. Our freely available implementation of the protocol is also described. It includes an LDAP server and a client library that makes writing LDAP programs much easier.

July 27, 1995

Center for Information Technology Integration
University of Michigan
519 West William Street
Ann Arbor, MI 48103-4943

Contents

h g c ec c h

- Author Information

[Next]

h g c ec c h

2. Overview of X.500

X.500 is the OSI directory service. X.500 defines the following components:

- An information model-determines the form and character of information in the directory.

- A namespace-allows the information to be referenced and organized.

- A functional model-determines what operations can be performed on the information.

- An authentication framework-allows information in the directory to be secured.

- A distributed operation model-determines how data is distributed and how operations are carried out.

The information model is centered around *entries*, which are composed of *attributes*. Each attribute has a *type* and one or more *values*. The type determines the attribute's *syntax*, which defines what kind of information is allowed in the values.

Which attributes are required and allowed in an entry are controlled by a special *objectClass* attribute in every entry. The values of this attribute identify the type of entry (e.g., person, organization, etc.). The type of entry determines which attributes are required, and which are optional. For example, the object class *person* requires the *surname* and *commonName* attributes, but description, *seeAlso*, and others are optional.

Entries are arranged in a tree structure and divided among servers in a geographical and organizational distribution. Entries are named according to their position in this hierarchy by a distinguished name (DN). Each component of the DN is called a relative distinguished name (RDN). Alias entries, which point to other entries, are allowed, circumventing the hierarchy. Figure 1 depicts the relationship between entries, attributes, and values and shows how entries are arranged into a tree.

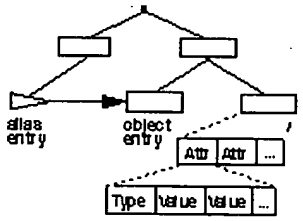


Figure 1. X.500 information model

The X.500 model is centered around entries composed of attributes that have a type and one or more values. Entries are organized in a tree structure. Alias entries can be used to build non-hierarchical relationships.

Functionally, X.500 defines operations in three areas: search and read, modify, and authenticate. In the first category, the *read* operation retrieves the attributes of an entry whose name is known. The list operation enumerates the children of a given entry. The *search* operation selects entries from a defined area of the tree based on some selection criteria known as a search filter. For each matching entry, a requested set of attributes (with or without values) is returned. The searched entries can span a single entry, an entry's children, or an entire subtree. Alias entries can be followed automatically during a search, even if they cross server boundaries.

In the second category, X.500 defines four operations for modifying the directory. The *modify* operation is used to change existing entries. It allows attributes and values to be added and deleted. The add and delete operations are used to insert and remove entries from the directory. The *modify RDN* operation is

h g c ec c h

used to change the name of an entry.

The final category defines a *bind* operation, allowing a client to initiate a session and prove its identity to the directory. Several authentication methods are supported, from simple clear-text password to public key-based authentication. The *unbind* operation is used to terminate a directory session. An *abandon* operation is also defined, allowing an operation in progress to be canceled.

Each X.500 operation and result can be *signed* to ensure its integrity. Signing is done using the originating client's or server's public key. The signed request or result is carried end-to-end in the protocol, allowing integrity to be checked at every step. This guards against connection hijacking or modification by intermediate servers. *Service controls* can be specified that determine information such as how an operation will be carried out, whether aliases should be dereferenced, the maximum number of entries to return, and the maximum amount of time to spend on an operation.

In X.500, the directory is distributed among many servers (called DSAs for Directory System Agent). No matter which server a client connects to, it sees the same view of the directory. If a server is unable to answer a client's request, it can either *chain* the request to another server, or *refer* the client to the server.

[Contents] [Previous] [Next]

h g c ec c h

3. Overview of LDAP

LDAP assumes the same information model and namespace as X.500. It is also client-server based, with one important difference: there are no referrals returned in LDAP. An LDAP server must return only results or errors to a client. If referrals are involved, the LDAP server is responsible for chasing them down. This model is depicted in Figure 2, though the intermediate server shown is not required (i.e., an implementation could choose to have its DSA speak "native" LDAP).

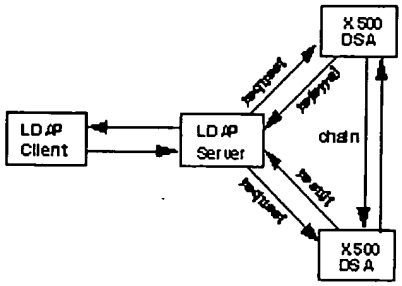


Figure 2. Relationship between LDAP and X.500

The LDAP client-server model includes an LDAP server translating LDAP requests into X.500 requests, chasing X.500 referrals, and returning results to the client.

The LDAP functional model is a subset of the X.500 model. LDAP supports the following operations: search, add, delete, modify, modify RDN, bind, unbind, and abandon. The search operation is similar to its DAP counterpart. A base object and scope are specified, determining which portion of the tree to search. A filter specifies the entries within the scope to select. The LDAP search filter offers the same functionality as the one in DAP but is encoded in a simpler form.

The time and size limit service controls are included directly in the search request. (They are not included with the other operations.) The *searchAliases* search control and *dereferenceAliases* service control are combined in a single *derefAliases* parameter in the LDAP search. The ASN.1 [11] definition of the LDAP search request is shown in Figure 3.

```
SearchRequest ::= [APPLICATION 3] SEQUENCE {
baseObject LDAPDN,
scope ENUMERATED {
baseObject (0),
singleLevel (1),
wholeSubtree (2)
},
derefAliases ENUMERATED {
neverDerefAliases (0),
derefInSearching (1),
derefFindingBaseObj (2),
alwaysDerefAliases (3)
},
sizeLimit INTEGER (0 .. MaxInt),
timeLimit INTEGER (0 .. MaxInt),
attrsOnly BOOLEAN,
filter Filter,
attributes SEQUENCE OF AttributeType
}
Filter ::= CHOICE {
and [0] SET OF Filter,
```

```
or [1] SET OF Filter,
not [2] Filter,
equalityMatch [3] AttributeValueAssertion,
substrings [4] SubstringFilter,
greaterOrEqual [5] AttributeValueAssertion,
lessOrEqual [6] AttributeValueAssertion,
present [7] AttributeType,
approxMatch [8] AttributeValueAssertion
```

Figure 3. ASN.1 for the LDAP search operation

The LDAP search operation offers similar functionality to DAP search. It combines search parameters and service controls and simplifies the filter encoding.

The *LDAPDN* and *AttributeType* components of the search are encoded as simple character strings using the formats defined in RFC 1779 [5] and RFC 1778 [2], respectively, rather than the highly structured encoding used by X.500. Similarly, the value in an *AttributeValueAssertion* is encoded as a primitive OCTETSTRING, not a more structured ASN.1 type. The structure is reflected in the syntax of the encoded string, not in the encoding itself.

The results of an LDAP search are sent back to the client one at a time, in separate *searchEntry* packets. This sequence of entries is terminated by a final *searchResult* packet that contains the result of the search (e.g., success, a size or time limit was exceeded, etc.). Having a final terminator packet allows clients and servers to *stream* results more easily, handling one entry at a time. This is especially useful in memory-constrained environments where holding the collection of all entries from a large search is not possible.

The X.500 list and read operations are not included in LDAP. Instead, they are emulated with the LDAP search operation. Read is emulated by a base object search of the entry to read, with a filter testing for the existence of the *objectClass* attribute. Every entry is required to have an object class and must match this filter. List is emulated by a one level search of the entry to list, also with a filter testing for the existence of the *objectClass* attribute. If the ability to distinguish alias children from other children (a feature provided by X.500 list) is desired, the *objectClass* attribute can be retrieved and examined for a value of alias.

The LDAP modify operation also differs slightly from its DAP counterpart. In DAP, four kinds of changes can be made: entire attributes can be added or deleted; individual values can be added or deleted. These capabilities require a client to read an entry before attempting a modify (e.g., when adding a value, to discover whether an *add attribute* or *add value* is required).

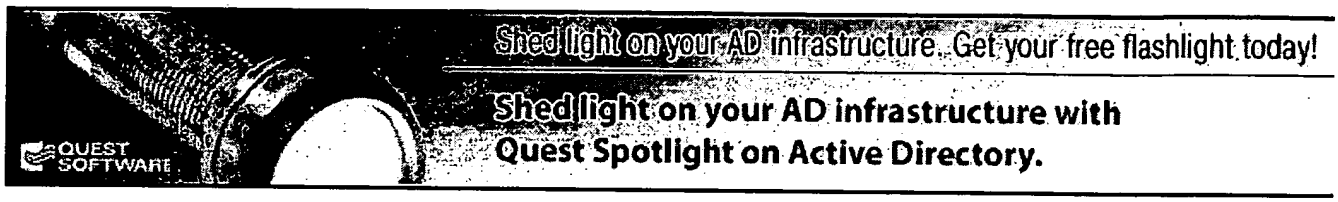
In LDAP, we simplified the semantics of modify by supporting three operations: add values; delete values; and replace values. If a request is made to add values to an attribute that does not exist in the entry, the attribute is created automatically. If a request is made to delete the last value of an attribute, the entire attribute is deleted. An attribute can also be deleted by specifying a *delete values* operation without specifying any values. Finally, the *replace values* construct is used to make an attribute contain the given values after the modify. The LDAP server takes care of translating the replace request into the necessary sequence of modify, add, and delete operations required by X.500.

The LDAP bind operation supports a subset of X.500 bind functionality. It allows only simple authentication, consisting of a clear-text password, and Kerberos version 4 authentication [6], which translates into an X.500 external authentication method. The LDAP bind operation includes a choice of credentials, allowing for future expansion of available authentication methods.

h g c ec c h

The DAP unbind, abandon, modify RDN, add and delete operations are virtually identical to their DAP counterparts.

[Contents] [Previous] [Next]



[Feature]

Accessing the Directory Database Using Distinguished Names

William Wong
InstantDoc #4901
March 1, 1999

A distinguished name (DN) is a text representation of an entry in the directory server database. A DN is a list of items, separated by commas, where each item is an attribute name followed by an equals sign character (=) and a value. For example, cn=Bill Wong, ou=Win NT Labs, c=US specifies a database entry with a common name (cn) of Bill Wong in an organizational unit (ou) of Win NT Labs in the country (c) of the United States (US). You use a DN to access entries in the directory database. Every item in the database has a unique DN, including the directory server and any other servers that it communicates with (e.g., the certificate server).

When you install and configure a directory server, you must specify several DNs. These entries include the root that will be common to all entries in the database, the directory server, the original administrator, and any other servers the administrator uses during the directory server installation, such as the certificate server or any replica directory servers.